

*Machine Translation in the Americas (AMTA-2006). – P. 223–231, Cambridge, Massachusetts.*  
19. Leusch, G., N. Ueffing and H. Ney. 2006. CDER: Efficient MT Evaluation Using Block Movements // *Proc. of the 11th Conference of the European Chapter of the Association for Computational Linguistics (EACL 2006).*  
20. Liu D., Gildea D. 2005. Syntactic Features for Evaluation of Machine Translation // *Proc.s of ACL Workshop on Intrinsic and Extrinsic Evaluation Measures for Machine Translation and/or Summarization.* – P. 25–32.

УДК 004.052; 519.62

П. Сердюк, О. Том'як

Національний університет "Львівська політехніка",  
кафедра програмного забезпечення

## АРХІТЕКТУРА ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ ВЕРИФІКАЦІЇ ТА АВТОМАТИЗОВАНОГО ТЕСТУВАННЯ МЕТОДІВ РОЗВ'ЯЗУВАННЯ ЗАДАЧ МАТЕМАТИЧНОЇ ФІЗИКИ

© Сердюк П., Том'як О., 2013

Розглянуто архітектуру програмного забезпечення для тестування та верифікації методів розв'язування систем рівнянь у часткових похідних, що описують фізичні процеси. Розроблено архітектуру процесора автоматизованого генерування задач, їх розв'язування та порівняльного аналізу з іншими методами аналізу задач математичної фізики. Проаналізовано роботу програмного забезпечення на оболонці COMSOL 4.2.

**Ключові слова:** автоматизоване тестування, диференціальні рівняння, математична фізика, верифікація.

**The article deals with software architecture for testing and verification methods for solving systems of equations in partial derivatives describing physical processes. Developed automated generation of processor architecture problems, their solution and comparative analysis with other methods of analysis tasks mathematical physics. Analyzed with software on shell COMSOL 4.2.**

**Key words:** automated testing, differential equations, mathematical physics, verification.

### Вступ

Важливим аспектом розроблення нових методів розв'язування систем диференціальних рівнянь у часткових похідних є їх верифікація та перевірка. Однією з найскладніших проблем автоматизованого тестування є покриття тестовими випадками різного роду моделей. У цій статті пропонується автоматизоване генерування тестових моделей для верифікації математичних методів розв'язування задач математичної фізики.

Для верифікації та визначення збіжності методів розв'язування задач математичної фізики використовують послідовну схему генерації пакетів для розроблення тестових шаблонів послідовного ланцюга. Така техніка значно збільшує кількість можливих завдань і зменшує надмірну специфікацію змінних стану, що іноді виникає у разі використання стандартного алгоритму [1, 2].

Сьогодні розробляється багато методів для розв'язування задач математичної фізики та початково-крайових задач для лінійних і нелінійних інтегральних рівнянь часткових похідних [3, 4]. Для верифікації, аналізу збіжності та стійкості розв'язків у статті пропонується використати емпіричний метод вимірювання характеристик методу, який полягає в автоматизованому генеруванні вхідних даних для методів розв'язування та їх подальшого аналізу. У статті розглянуто архітектуру програмного забезпечення, яке складається з: трансляторів у математичні скриптові мови, генератора тестових конструкцій та модуля верифікації.

Задача генерування тестових конструкцій полягає у наданні вхідних даних за відповідними обмеженнями: просторовими, обмеженнями на фізичні величини та конструкції. Генерування

складних конструкцій та їх обробка розглянуті в [5], де представлено метод обробки поверхонь, що генеруються як розв'язок рівняння часткових похідних. Використовується двовимірний параметр, з якого визначається проекція обробленої кривої на представлену параметрично поверхню. Для визначеної обробленої кривої задають граничні умови, що дають змогу розв'язувати еліптичні рівняння в параметричному просторі. В методі генерації поверхонь на основі еліптичних рівнянь часткових похідних граничні умови використовуються для контролю форми поверхні. Це дає змогу отримати поверхні в замкнутій формі, навіть для випадку загальних граничних умов. У [6] представлено комплексний підхід, який може передбачати генерування поверхні на основі моделі потенціалів, завдяки чому можливий інтерактивний дизайн гнучких топологічних поверхонь і модифікація узагальнених граничних умов, а також накладання різних геометричних і фізичних обмежень, що дає змогу підтримувати різні інтерактивні методи та контроль над границями.

Отже, задачі верифікації та аналізу результатів методів задач математичної фізики актуальні, і поки що немає загального підходу до тестування та верифікації методів розв'язування задач математичної фізики за допомогою емпіричних автоматизованих досліджень. У статті розглянуто розроблене програмне забезпечення для автоматизованого генерування тестових моделей та подальшої перевірки визначення збіжності, стійкості та інших характеристик методів розв'язування задач математичної фізики.

## 2. Архітектура генерування тестових задач математичної фізики

Для того, щоб забезпечити гнучкість створення тестових задач під різними обчислювальними математичними платформами, розроблена архітектура повинна мати такі властивості:

- легкість тестування нових методів розв'язування систем диференціальних рівнянь часткових похідних;
- легкість під'єднання системи до нових обчислювальних оболонок, що використовують сучасні математичні скриптові мови;
- цілісність та структурованість даних для подальшої верифікації та постобробки результатів.

З цією метою у розроблену архітектуру введено три компоненти: генератор тестових завдань, конвертор вхідних даних та модуль верифікації результатів. Вихідні результати задаються масивом точок та відповідним значенням розв'язку.

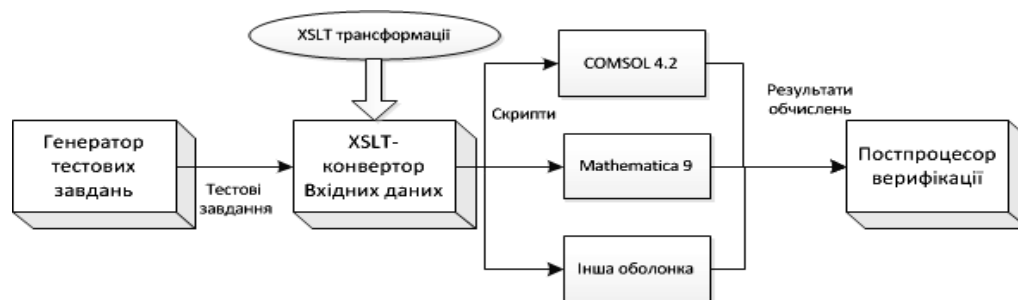


Рис. 1. Базова архітектура програмного забезпечення тестування та верифікації методів розв'язування задач математичної фізики

Вхідні тестові задачі можуть задаватися на рівні клієнта або надходити з системи генерування з відповідними обмеженнями. Алгоритм автоматизованого генерування тестових задач створює вхідні математичні моделі, а саме:

- конструкції та параметри граничних та початкових умов;
- диференціальні рівняння в частинних похідних, які описують фізичні процеси.

За допомогою XSLT трансформацій дані структури переводимо у код відповідної оболонки. Для тестування нового методу розв'язування задач у часткових похідних необхідно задати відповідні XSLT перетворення, щоб отримати вхідні дані для математичних оболонок.

Постпроцесор верифікації перевіряє коректність отриманого розв'язку, а також досліджує збіжність та стійкість, похибку одержаних розв'язків залежно від параметрів моделі.

Опис тестових структур вхідних даних містить класи, які дають змогу гнучко представляти різні види широкого класу задач. У зв'язку з великою кількістю структурних елементів розглянемо лише деякі із них, подані на рис. 1.

Клас PDETask (рис. 1) призначений для задання умови диференціального рівняння і є контейнером списку областей та списку граничних та початкових умов. Основна вимога до областей – це їх зв'язність та неперервність. Клас Boundary призначений для збереження граничних умов, що задають поведінку рівняння на границі розглянутої області, яка визначається у класі Domain та містить такі дані: назва інтервалу, простір, тип і список змінних. Поверхню задають за допомогою параметричних рівнянь гомеоморфного відображення одиничного квадрата (або лінії чи точки, залежно від просторового виміру). У багатьох математичних оболонках немає можливості такого представлення, тому генератор містить алгоритм тріангуляції поверхні, який застосовується при трансляції у математичні скрипти. Між цими структурами існує зв'язок композиції, оскільки граничні умови та інтервали є складовими умови рівняння.

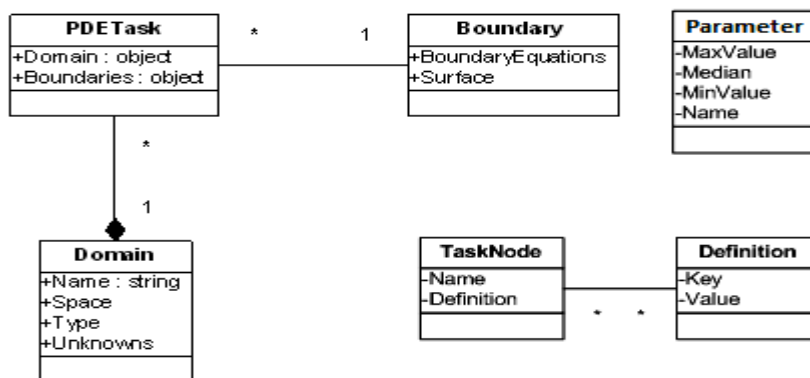


Рис. 2. Основні класи вхідної задачі

Розглянемо генерування тестових задач на прикладі параметризації граничних умов. Граничні умови задають за допомогою загального диференціального рівняння на границі. Особливим випадком є граничні умови третього роду, або граничні умови Робена, які задають у вигляді лінійної комбінації значення шуканої функції та її похідної, які легко параметризуються для генерування тестових задач:

$$a y'(a) + b y(a) + g = 0. \tag{1}$$

Для граничних умов Робена генератор змінюватиме  $a$  та  $b$  у заданих наперед інтервалах, генеруючи різні задачі, у результаті розв'язання яких можна досліджувати вплив. Для граничних умов у загальному випадку параметри, які може змінювати генератор, необхідно задавати вручну.

У випадку опису поверхонь, що задаються складнішими рівняннями (гіперболічними, параболічними тощо) доводиться дотримуватись ряду обмежень.

Враховуючи всі вимоги, розроблено архітектуру для програмного забезпечення верифікації та автоматизованого тестування методів розв'язування диференціальних рівнянь. Ця архітектура уможливіє легку адаптацію програмного засобу до різних математичних обчислювальних платформ, забезпечує структурованість та цілісність даних для коректної роботи програмного засобу.

### 3. Результати роботи генератора

Для тестування алгоритму використано поширену конструкцію кристала МЕН, встановленого на жорсткі виводи. Досліджувана конструкція складається з алюмінієвого тепловідводу (розміру  $9 \times 140 \times 10$  мм) та встановленого на ньому на чотирьох алюмінієвих виводах ( $4 \times 4 \times 10,5$  мм) кремнієвого кристала  $\text{SiO}_2$  ( $50 \times 100 \times 10$  мм), простір між якими заповнений полімером PVC. Джерела тепла розміщуються в центрі нижньої поверхні кристала, їх розміри становлять  $10 \times 5 \times 1$  мм,  $5 \times 10 \times 1$  мм,  $10 \times 10 \times 1$  мм. Вхідні дані підібрано так, щоб якомога повніше відобразити функціональність алгоритму візуалізації.

У результаті роботи програмного забезпечення згенеровано код за допомогою XSLT-перетворень, лістинг якого наведено нижче. Кожна з конструкцій додається за допомогою відповідного XSLT тегу.

Ліст. 1. Лістинг згенерованого коду для оболонки Comsol Script 4.2

```
import com.comsol.model.*
import com.comsol.model.util.*
model = ModelUtil.create('Model');
model.modelPath('D:/Generator/Sample/Simulation 1 - Therma');
model.modelNode.create('mod1');model.geom.create('geom1', 2); model.mesh.create('mesh1', 'geom1');
model.physics.create('emw', 'ThermalMechanical', 'geom1');
model.study.create('std1'); model.study('std1').feature.create('mode', 'ModeAnalysis');
model.geom('geom1').feature.create('r1', 'Block'); model.geom('geom1').run('r1');
model.geom('geom1').run('r1'); model.geom('geom1').feature.create('r2', 'Block');....
```

На рис. 3 зображено об'ємну візуалізацію розподілу температури у МЕП методом візуалізації перетинів, отриманих у результаті розв'язування задачі за допомогою COMSOL 4.2. На рис. 4 зображено розподіл температури.

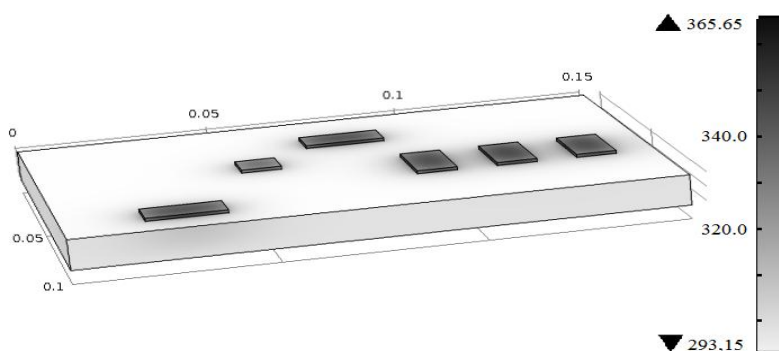


Рис. 3. Візуалізація розподілу температури у тестовій конструкції

Вихідні результати задаються масивом точок та відповідним значенням розв'язку.

### Висновок

Розроблено інтерактивний постпроцесор для генерування тестових задач математичної фізики з гнучкою архітектурою, який дає змогу верифікувати методи задач математичної фізики. Запропоновано новий підхід до верифікації та автоматизованого тестування методів розв'язування задач математичної фізики, у якому використано наявні алгоритми генерації тестових вхідних даних, модифікованої відповідно до області застосування. Для тестування нового методу розв'язування задач у часткових похідних необхідно задати відповідні XSLT перетворення для отримання вхідних даних. Вхідні тестові задачі можуть задаватися на рівні клієнта або надходити з системи генерування з відповідними обмеженнями. Алгоритм автоматизованого генерування тестових задач математичної фізики розглянемо у подальших роботах.

1. Niermann, Patel HITEC: *A test generation package for sequential circuits* / Niermann, Patel // *Design Automation. EDAC., Proceedings of the European Conference on.* – 1991. – P. 214–218. 2. Cooper, Graham, Sweller John. *Effects of schema acquisition and rule automation on mathematical problem-solving transfer* / Cooper, Graham, Sweller, John // *Journal of Educational Psychology.* – 1987. – P. 347–362. 3. Danping Peng, Barry Merriman, Stanley Osher, Hongkai Zhao, Myungjoo Kang. *A PDE-Based Fast Local Level Set Method* / Danping Peng, Barry Merriman, Stanley Osher, Hongkai Zhao, Myungjoo Kang // *Department of Mathematics, University of California at Los Angeles.* – 1999. – P. 410–438. 4. Fokas *A unified transform method for solving linear and certain nonlinear PDEs/ Department of Mathematics, Imperial College of Science, Technology and Medicine, University of London, London SW7 2BZ, UK, Institute of Nonlinear Studies, Clarkson*

University, Potsdam, NY. – 1997. – P. 1411–1443. 5. Sirendaorejia, Sun Jiong Auxiliary equation method for solving nonlinear partial differential equations / Department of Mathematics, Inner Mongolia Normal University, Huhhot 010022, Inner Mongolia, PR China. – 2003. – P. 387–396. 6. Haixia Du, Hong Qin Dynamic PDE-based surface design using geometric and physical constraints/ Department of Computer Science, State University of New York at Stony Brook, Stony Brook, NY. – 2005.– P. 43–71.

УДК 004

Д. Махун, Б. Демида  
Національний університет “Львівська політехніка”,  
кафедра автоматизованих систем управління

## ДОСЛІДЖЕННЯ ТЕХНОЛОГІЇ СТВОРЕННЯ ПРИКЛАДНИХ АПЛІКАЦІЙ ДЛЯ ОС ANDROID

© Махун Д., Демида Б., 2013

Досліджено технології створення для операційної системи мобільних платформ Android аплікації графічного представлення динамічно-змінної інформації. Наведено приклад розв’язання поставленої задачі відтворення розкладу для студентів/викладачів Національного університету “Львівська політехніка” з використанням засобів Android SDK. Аплікація є графічним інтерфейсом, що отримує дані від веб-ресурсу по протоколу HTTP за допомогою безпроводної мережі.

**Ключові слова:** аплікація, Android, Java, Google, Android SDK, мобільний маркет, програмування на смартфоні, розроблення аплікацій, мобільні платформи.

**In this article investigated methods of graphic representation for dynamically changed information – namely, representation for students'/lecturers' schedule of Lviv Polytechnic National University – in an application for operating system Android. Given example of solving current task using utilities of Android SDK. Android application is a graphical interface that receives data from a Web-resource (site) using wireless technology for data networks – web service technology. The relation between web-site and application is based on HTTP-requests on the Internet.**

**Key words:** application, Android, Java, Google, Android SDK, mobile market, smartphones programming, application development, mobile platforms.

### Аналіз предметної області

В сучасному комп’ютеризованому світі за динамічного розвитку у всіх сферах суспільства надзвичайно важливим є своєчасне володіння достовірною інформацією. Найкраще із цією функцією справляється технологія WWW, яка дає змогу пов’язувати в одній мережі різнопланову інформацію і представити її у вигляді веб-сторінок. З появою, поширенням мобільних пристроїв та розвитком веб-технологій стало можливим отримувати інформацію будь-де через Інтернет та зберігати її для подальшого використання. Операційна система для смартфонів Android забезпечує отримання та відображення інформації з веб-ресурсів у зручний для користувача спосіб.

### Постановка задачі та формулювання мети дослідження

**Головною метою** цього дослідження є застосування сучасних досягнень у веб-технологіях для створення програмного продукту – аплікації для мобільної системи Android, яка дає змогу швидко відображати структурований набір даних, наприклад, розклад викладача або студента університету. Вже розроблено веб-ресурс для відображення розкладу для університету, проте для того, щоб отримати розклад, користувач змушений здійснювати цілу низку переходів по різних веб-сторінках та обов’язково бути підключеним до Інтернету. Такі обмеження викликають незручності у користуванні розкладом і тим самим зменшують потік потенційних користувачів.